Technical Note: Fast Unitary Mixing with Data-Dependent Rotations and Energy Routing

1) Core idea

Construct layers by alternating:

$x \to H \to R(x) \to H \to R(x) \to \cdots$

- H: fixed global mixing (e.g. Fast Walsh–Hadamard Transform), ($O(n \log n)$)
- R(x): **pairwise unitary rotations** with **data-dependent angles**

Each step is (locally) orthogonal:

- preserves norms
- well-conditioned
- supports deep stacking

---

2) Pairwise rotation layer

Basic unit (per pair i, j):

$$\begin{bmatrix} x_i' \\ x_j' \end{bmatrix} = \begin{bmatrix} \cos\theta_{ij}(x) & -\sin\theta_{ij}(x) \\ \sin\theta_{ij}(x) & \cos\theta_{ij}(x) \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix}$$

Properties:

- Orthogonal for any θ
- Cost: O(n) per layer
- Entire layer = block-diagonal matrix of 2×2 rotations

---

3) Why alternate with H

- Pairwise rotations alone → only local mixing
- H → global mixing (every coordinate influences every other)

Effect over layers:

- local rotations + global reshuffling → full dense interaction
- different pairings are implicitly realized after each H

---

4) Data-dependent angle generation (key design space)

The choice of how $\theta_{ij}(x)$ is computed is *central*.

A. Local angle functions

- $\theta_{ij} = f(x_i, x_j)$
- Examples:
  - linear: $ax_i + bx_j$
  - bilinear: ax_ix_j
  - small MLP on (x_i, x_j)

Pros:

- cheap
- easy to scale
- preserves locality

Cons:

- limited global awareness (before mixing)

B. Globally conditioned angles

- $\theta = g(Hx)$, where $g: R^n \rightarrow R^{n/2}$

Pipeline:

- compute u = Hx
- generate all angles from u
- apply pairwise rotations

Pros:

- each rotation sees global context
- more expressive

Cons:

- higher cost
- more parameters

C. Hybrid

- $\theta_{ij}=f(x_i,x_j,u_i,u_j)$, with u=Hx **Interpretation:**
- local signal + globally mixed signal

D. Static learned angles

- θ_{ij} are parameters (not data-dependent)

Pros:

- simplest
- stable

Cons:

- linear transform overall
- limited adaptability

---

5) Angle parameterization

Avoid direct angle optimization when possible.

Options:

- **Options:**
  - **Normalized sin/cos:**
    - parameters a, b
    - $c = a/\sqrt{a^2+b^2},\ s = b/\sqrt{a^2+b^2}$
  - **Bounded angle:**
    - $\theta = \alpha \tanh(h(x))$
  - **Small-angle initialization:**
    - initialize near identity ($\theta \approx 0$)

---

6) Pairing strategies

Fixed pairing

- (1,2), (3,4), …
- simplest

Layer-wise reshuffling

- permute indices between layers

Implicit reshuffling via H

- often sufficient:
  - H redistributes coordinates globally
  - effective pairings change each layer

---

7) Energy routing via width expansion

Construction:

- expand: $x \in \mathbb{R}^n \to \tilde{x} \in \mathbb{R}^{2n}$
- apply unitary stack
- read out only part of the output

$$\tilde{x} \xrightarrow{U(x)} z = (z_{\text{visible}}, z_{\text{hidden}})$$

Identity:

$$\|\tilde{x}\|^2 = \|z_{\text{visible}}\|^2 + \|z_{\text{hidden}}\|^2$$

Interpretation:

- rotations control **energy transfer between subspaces**
- visible channels = signal
- hidden channels = sink / buffer

---

8) What energy routing gives you

- indirect amplitude modulation:
  - keep energy → strong output
  - move energy to hidden → suppress output
- feature selection without breaking orthogonality
- stable gradients (no scaling inside the transform)

---

9) Practical controls for energy routing

- penalty on hidden energy:
  - encourages use of visible channels
- normalization on visible output
- adjustable split (not necessarily 50/50)
- multiple hidden partitions (hierarchical routing)

---

10) Computational profile

- H: O(nlogn)
- rotations: O(n)
- total per layer: O(nlog n)

Memory:

- minimal parameters (angles only)
- no dense matrices

---

11) Conceptual summary

This architecture separates three roles:

- **Global mixing** → via fast transforms
- **Adaptive interaction** → via data-dependent rotations
- **Amplitude control** → via energy routing between subspaces

---

12) Minimal mental model

- Think of H as a **global shuffle**
- Think of rotations as **local, input-controlled mixers**
- Think of width expansion as **giving the system somewhere to send unwanted energy**

---

13) Key design knobs (quick checklist)

- angle generator: local / global / hybrid
- angle parameterization: sin–cos vs bounded
- pairing: fixed vs reshuffled
- width expansion: none / 2× / multi-partition
- energy regularization: yes / no